VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Základní metody rozpoznávání akcí

Basic Methods for Action Recognition

2017 Jan Ožana

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Jan Ožana**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: Základní metody rozpoznávání akcí
Basic Methods for Action Recognition

Jazyk vypracování: čeština

Zásady pro vypracování:

Rozpoznávání akcí a činností je v poslední době věnována značná pozornost výzkumu. Úspěšné metody mohou být využity v mnoha aplikacích, jako je bezpečnost, analýza lidského chování apod. Je proto potřeba vyvíjet spolehlivé metody pracující v reálném čase.

1. Seznamte se se základními technikami detekce pohybu ve videosekvencích.
2. Seznamte se s metodami detekce a klasifikace akcí.
3. Vybrané metody popište.
4. Experimentálně ověřte funkčnost minimálně jednoho algoritmu.
5. Zjištěné poznatky řádně zdokumentujte v textu práce.

Seznam doporučené odborné literatury:

[1] Ronald Poppe: A survey on vision-based human action recognition, Image and Vision Computing, Volume 28, Issue 6, June 2010, Pages 976-990, ISSN 0262-8856
[2] Daniel Weinland, Remi Ronfard, Edmond Boyer: A survey of vision-based methods for action representation, segmentation and recognition, Computer Vision and Image Understanding, Volume 115, Issue 2, February 2011, Pages 224-241, ISSN 1077-3142

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Radek Simkanič, DiS**

Datum zadání: 01.09.2016

Datum odevzdání: 28.04.2017

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 28. dubna 2017 .....

Rád bych poděkoval vedoucímu práce Ing. Radku Simkaniči, DiS za trpělivost, odbornou pomoc a konzultaci při vytváření této bakalářské práce. Dále bych rád poděkoval všem, kteří mě po celou dobu studií podporovali.

Abstrakt

Tématem této bakalářské práce jsou základní metody rozpoznávání akcí. V práci jsou popisovány techniky detekce pohybu ve videosekvencích a metody detekce a klasifikace akcí. Následuje implementace vybrané metody, která je založena na trajektoriích pohybu jednotlivých kloubů člověka. K implementaci byl využit programovací jazyk C++ a knihovna pro práci s obrazem OpenCV.

Klíčová slova: Rozpoznávání akcí, klasifikace akcí, detekce pohybu, příznaky, OpenCV, C++, Kinect

Abstract

Theme of this bachelor's thesis are basic methods for action recognition. The thesis describes techniques of motion detection in video sequences and methods of action detection and classification. Next topic of the thesis is the implementation of selected method of action recognition. The method is based on joint trajectories. The work is based on programming language C++ and library of programming functions aimed at computer vision OpenCV.

Key Words: Action recognition, action classification, motion detection, features, OpenCV, C++, Kinect

Obsah

Seznam použitých zkratek a symbolů

CMOS	–	Complementary Metal–Oxide–Semiconductor - technologie obrazových čipů
CCD	–	Charge-Coupled Device - technologie obrazových čipů
RGB	–	Red Green Blue - barevný model
RGB-D	–	Red Green Blue - Depth - typ snímacího zařízení
SDK	–	Software Development Kit - sada vývojových nástrojů
2D	–	dvoudimenzionální
3D	–	trojdimenzionální
4D	–	čtyřdimenzionální
HoG	–	Histogram of Oriented Gradients - histogram orientovaných gradientů
STIP	–	Space-Time Interest Point
STOP	–	Space-Time Occupancy Pattern
ROP	–	Random Occupancy Patterns
LOP	–	Local Occupancy Patterns
FTP	–	Fourier Temporal Pyramid
SVM	–	Support Vector Machines - metoda strojového učení
XML	–	Extensible Markup Language - obecný značkovací jazyk
BSD	–	Berkeley Software Distribution - licence pro svobodný software
OpenCV	–	Open Source Computer Vision -knihovna pro manipulaci s obrazem.
CPU	–	Central Processing Unit - centrální procesorová jednotka
GPU	–	Graphic Processing Unit - grafický procesor
RAM	–	Random-access memory - druh počítačové paměti
SSD	–	Solid-state drive - typ datového média
OS	–	Operating System - operační systém

Seznam obrázků

Seznam tabulek

1 Úvod

Rozpoznávání akcí a činností se v poslední době stává velmi důležitým tématem počítačového vidění a je mu věnována značná pozornost výzkumu. Úspěšné metody mohou najít využití v širokém spektru aplikací, například robotice, bezpečnosti, analýze lidského chování, interakci mezi člověkem a strojem apod. Je proto potřeba vyvíjet spolehlivé metody pracující v reálném čase.

V rámci první části této bakalářské práce budou popsány nejběžnější možnosti snímání obrazu pro účely počítačového vidění. V této částí budou také přiblíženy základní techniky detekce pohybu ve videosekvencích, které jsou důležitou součástí algoritmů pro rozpoznávání akcí. Další kapitola se již bude zabývat přímo samotnými metodami rozpoznávání akcí prováděných člověkem. Budou zmíněny teoretická východiska, druhy prostorových reprezentací akcí a jednotlivé metody získávání příznaků z obrazu (STIP, STOP, ROP, LOP apod.). Na konci této kapitoly budou popsány metody strojového učení.

Poslední část práce se bude zabývat vlastní implementací vybrané metody. Cílem této části bude vyvinout algoritmus rozpoznávání akcí založený na trajektoriích pohybu jednotlivých kloubů člověka. Bude také popsáno jak hardwarové vybavení použité pro snímání sekvencí v datasetu, tak i samotný dataset, který bude využit. Následuje popis knihovny OpenCV, což je v současné době jedna z nejpoužívanějších knihoven pro práci s obrazem. Jednou z posledních částí práce je popis implementace algoritmu a jeho testování, při kterém bude brán ohled hlavně na správnost určení akcí. Na konec této části bude popsán výsledný program.

2 Počítačové vidění

Počítačové vidění je jednou z nejpokročilejších a nejsložitějších disciplín počítačové vědy a vývoje softwaru. Jedná se o snahu vytvořit technologii, která umožní strojům vidět a napodobit tak lidské vnímání okolního světa. Nejedná se pouze o zachycení obrazu různými elektronickými prostředky, jako jsou například CMOS nebo CCD senzory, ale také porozumění a rozpoznání objektů a akcí v obraze se vyskytujících pomocí softwaru.

Nedílnou součástí rozpoznávání a klasifikace akcí je také detekce pohybu ve videosekvencích. K tomu, aby mohlo začít rozpoznávání akcí, musí být nejdříve detekován pohyb. V případě aplikování algoritmů rozpoznávání na celé videosekvence by docházelo k zbytečnému vytěžování systémových prostředků zařízení, na kterém toto běží.

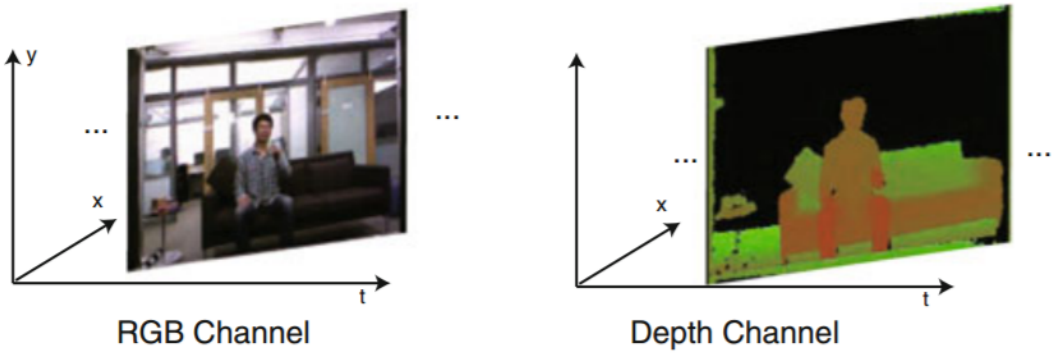
2.1 Snímání obrazu

Kamera je elektronické zařízení umožňující snímání pohyblivého obrazu. Klasické, běžně dostupné, digitální kamery obsahují snímač obrazu CCD či CMOS [1]. Oba druhy těchto obrazových snímačů jsou tvořeny určitým počtem světlocitlivých buněk, které při dopadu fotonů světla produkují elektrický náboj korespondující s intenzitou dopadajícího světla. Čím vyšší je intenzita světla dopadajícího na jednotlivé buňky snímače, tím větší elektrický náboj je produkován. Na základě velikosti těchto nábojů jsou následně dopočítány intenzity jasu v obraze a scéna je tak zrekonstruována v digitální podobě. Tento celý proces digitalizace obrazového signálu lze rozdělit do několika kroků:

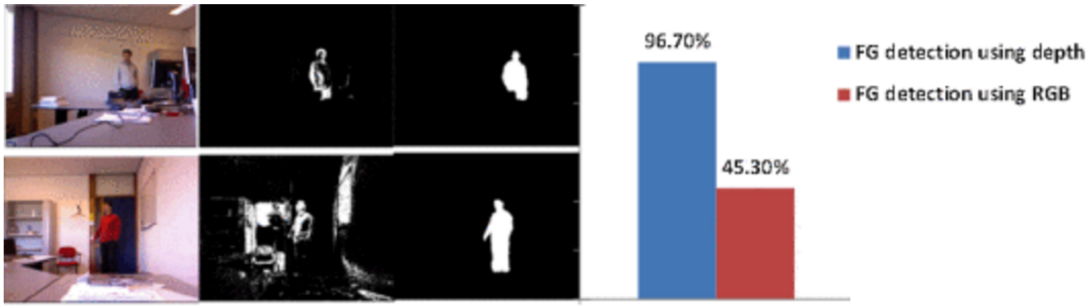
1. Přivedení reálného obrazu do kamery pomocí čoček, soustavy zrcadel, či optických vláken. Toto je zpravidla prováděno pomocí objektivu kamery.
2. V případě barevného obrazu je nutno rozdělit barevné složky obrazu pomocí vhodných optických filtrů.
3. Zachycení signálů prostřednictvím světlocitlivých senzorů.
4. Zachycení elektrických signálů z jednotlivých buněk senzoru.
5. Vytvoření digitální reprezentace obrazu na základě elektrického signálu.

Hloubkové kamery využívají strukturovaného světla pro snímání hloubkových map v reálném čase [2]. Na rozdíl od klasických kamer, které měří intenzitu světla, či barvy v obraze, hloubkové kamery zaznamenávají hloubku scény. Také výsledný obraz z obou typů kamer je velmi odlišný viz obrázek 1.

Využití hloubkových map má několik výhod oproti klasickým RGB obrazům konvenčních kamer. Například dobrá funkčnost i ve špatných světelných podmínkách, či nezávislost na barvě a textuře sledovaného objektu jsou podstatnými výhodami v prospěch dat získaných prostřednictvím hloubkových kamer. Naopak nevýhodami hloubkových kamer je výrazný šum v obraze



Obrázek 1: Porovnání klasického RGB snímku se snímkem získaným z hloubkové kamery [19]



Obrázek 2: Porovnání úspěšnosti detekce popředí snímku při použití RGB kamery a hloubkové kamery. Zleva: RGB snímek scény, maska popředí za využití RGB dat, maska popředí za využití hloubkových dat. Modrý sloupec grafu znázorňuje úspěšnost detekce popředí za využití hloubkových map, červený sloupec grafu znázorňuje úspěšnost detekce popředí za využití RGB dat. [2]

a degradace přesnosti se zvyšující se vzdáleností pozorovaného objektu od kamery. V případě použití pouze jedné hloubkové kamery také může dojít k problematické rozlišitelnosti pozadí a popředí snímku.

2.2 Analýza pohybu

V rámci rozpoznávání akcí by bylo neefektivní aplikovat algoritmy rozpoznávání na celou sekvenci snímků, proto je vhodné využít různých algoritmů na detekování pohybu v sekvencích a až poté aplikovat samotné algoritmy rozpoznávání akcí. V současné době se využívá k detekci pohybu ve videosekvencích široké spektrum algoritmů či metod [3]. Jednou ze základních metod je takzvaný odečet pozadí, anglicky též background substratction. Jde o jednu z nejjednodušších metod, jejíž princip je založen na vzájemném porovnávání referenčního snímku pozadí, na kterém se nevyskytuje žádný objekt, a aktuálního snímku. Tato metoda však neposkytuje vždy dostačující výsledky, neboť pozadí snímků nemusí být vždy stoprocentně statické, a tudíž algo-

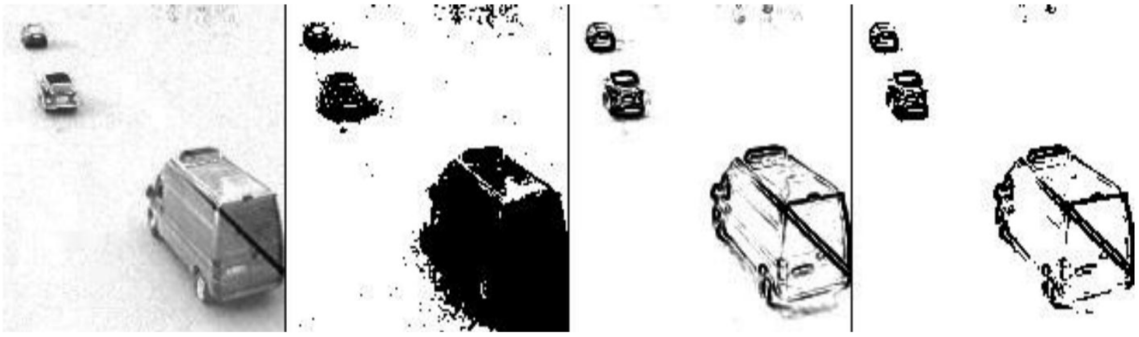
Obrázek 3: Příklad fungování detekce pohybu na základě odečtu pozadí. Zleva: referenční snímek pozadí, aktuální snímek, snímek s odečteným pozadím. [4]

ritmus může detekovat pohyb i v případech, kdy to není žádoucí. Jde například o změny šumu v obraze produkovaného kamerou, změny osvětlení, pohyb korun stromů vlivem větru apod. Ve videosekvenci lze samozřejmě vymezit části obrazu, které nejsou sledovány a na případný pohyb v nich tak nebude brán zřetel. Existují také ale i jiné metody, než je odečet pozadí, například:

- porovnání histogramu mezi snímky,
- sledování rozdílných bodů mezi snímky,
- porovnání jednotlivých snímků zpracovaných detektorem hran,
- metoda založená na optickém toku v obraze.

2.2.1 Porovnání histogramů mezi snímky

Mezi další ze základních metod detekce pohybu v obraze patří metoda detekce založená na porovnání histogramů mezi snímky [3]. Je založena na porovnávání světelné charakteristiky (histogramů hodnot jasů pixelů) jednotlivých snímků sekvence se vzorovým snímkem obsahujícím pouze pozadí bez jakéhokoliv pohybujícího se objektu. Pokud dojde ke změně světelné charakteristiky, která překročí předem nastavený práh, znamená to, že byl právě detekován pohyb ve scéně. Pokud by nebyl použit práh, docházelo by k častému falešnému detekování pohybu. Avšak i při použití prahu může dojít k nesprávnému vyhodnocení, což může být zapříčiněno hlavně změnami v prostředí, které nelze předvídat (změna počasí, rozestavení scény apod.). Toto lze částečně omezit častou aktualizací vzorového snímku pozadí. Další nevýhodou této metody může být neschopnost určit místo pohybu ve scéně v případě porovnávání histogramů celých snímků. V případě rozdělení snímku na více podčástí lze docílit i určení místa pohybu ve scéně. I přes zmiňované nevýhody se stále jedná o jednu z nejpoužívanějších a nejrozšířenějších metod detekce pohybu v obraze. Hlavní výhodou této metody je její jednoduchost a výpočetní nenáročnost.



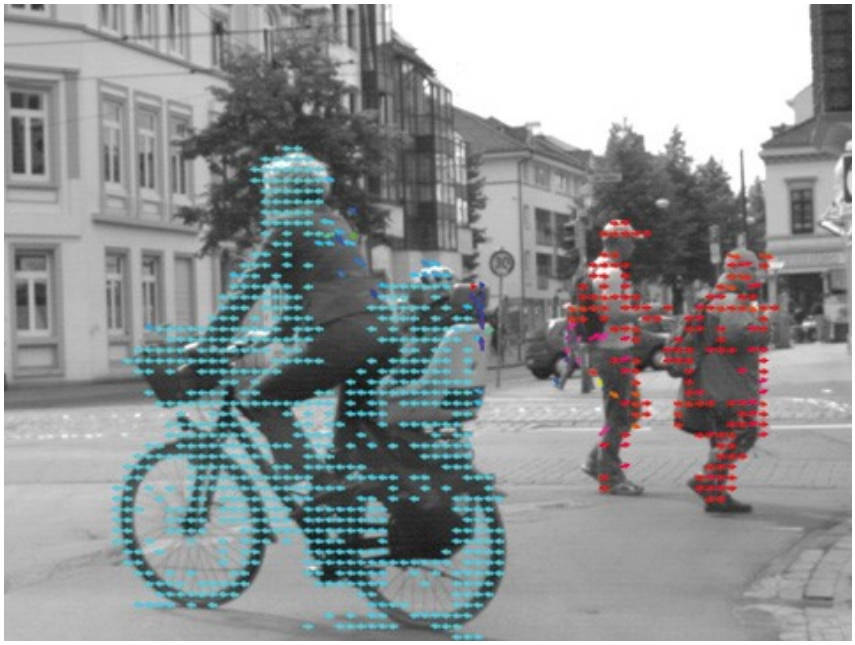
Obrázek 4: Porovnání snímků zpracovaných detektorem hran. Zleva: původní rozdílový snímek, původní snímek po prahování, původní snímek po aplikaci algoritmu pro zvýraznění hran, snímek se zvýrazněnými hranami po prahování [3]

2.2.2 Sledování rozdílných bodů mezi snímky

Jedná se o jednu z dalších základních metod detekce pohybu v obraze. Tato metoda je založena na principu porovnávání korespondujících pixelů mezi dvěma snímky sekvence [3]. Pokud nastane situace, kdy je výraznější rozdíl mezi hodnotami jednotlivých pixelů, byl detekován pohyb. V rámci této metody je ideálním přístupem využití histogramů, kdy jeden ze dvou histogramů je vždy již předpočítán z předchozího kroku. V praxi by však tato metoda ve své základní podobě nefungovala příliš účinně, a to hlavně z důvodu vysoké výpočetní náročnosti v případě práce s každým pixelem obou snímků. Snížení výpočetní náročnosti může být dosaženo převodem barevného snímku v RGB modelu do odstínů šedi. Touto úpravou neztratí sekvence svou vypovídací hodnotu o pohybu v ní obsaženém. Podstatnou výhodou této metody je její schopnost přesně lokalizovat pohyb v obraze a v případě stálého pozadí také schopnost detekovat malé, pomalu se pohybující objekty. Jednou z hlavních nevýhod metody v základní podobě je její výpočetní náročnost.

2.2.3 Porovnání snímků zpracovaných detektorem hran

Metoda porovnání snímků zpracovaných detektorem hran je založena na použití algoritmu, který je určen pro zvýraznění hran v obraze před prahováním (prahování je jedna z metod segmentace obrazu, kdy všechny pixely s hodnotou nižší než práh jsou detekovány jako pozadí a všechny pixely s hodnotou vyšší než práh jsou detekovány jako popředí) [3]. Hrany, jakožto kraje objektů, poskytují dostatek informací k porovnání snímků. Hlavní výhodou využití detektoru hran v rámci algoritmu detekce pohybu je zbavení se drobného šumu, který vzniká na snímači kamery. Pokud šum v obraze netvoří souvislou oblast, nejsou v něm tedy detekovány žádné hrany, je ignorován. Teprve na takto upravené snímky se využívají další metody detekce pohybu, například již výše zmíněné metody porovnání histogramu mezi snímky nebo sledování rozdílných bodů mezi snímky.



Obrázek 5: Ukázka optického toku v obraze, barevně jsou zvýrazněny pohybující se objekty. [5]

2.2.4 Metoda založená na optickém toku

Metoda založená na optickém toku využívá k detekci pohybu sledování jednotlivých pixelů ve videosekvenci [3]. Lze tak určit přesný pohyb objektu v obraze, stejně jako rychlost tohoto pohybu. Tato metoda je však výpočetně velmi náročná, a tudíž jí nelze aplikovat na každý pixel v obraze. Proto lze určit pouze několik bodů v obraze, které budou trvale sledovány, a až v případě detekce pohybu bude spuštěn výpočetně náročnější algoritmus, který bude aplikován na vyšší počet bodů v obraze.

3 Metody rozpoznávání akcí

Rozpoznávání akcí je v posledních letech žhavým tématem oboru počítačového vidění, kterého může být využito v celé řadě lidských činností, ať už jde o interakci počítače s člověkem, ovládání strojů, bezpečnost, či analýza lidského chování [6]. Rozpoznávání akcí je proces, při kterém dochází k pojmenování akce prováděné člověkem na základě různých algoritmů. Vstupem do těchto algoritmů, na základě kterých dochází k pojmenování akcí, může být například video či více videí, data z různých senzorů, nebo také zvuková stopa. Tato práce se však bude zabývat pouze metodami využívající obraz.

Při vývoji algoritmů a metod rozpoznávání akcí však narážíme na několik problémů. Tyto metody musí být ideálně univerzálně použitelné na jakoukoliv osobu, která akce provádí v jakémkoliv prostředí. Musí být nezávislé na postavě člověka, jeho oblečení, věku, gestech, rychlosti provedení akce apod. Některé algoritmy mohou být také nezávislé na postavení kamery vůči sledované osobě.

3.1 Extrakce příznaků

Je třeba definovat, co je myšleno slovem příznak. Lze takto nazývat cokoliv, co má ve videosekvenci nějakou užitečnou informaci pro další využití, například určitý tvar, barvu, texturu. Avšak jako příznak lze označit také například lidský obličej, celý dům, registrační značku vozidla či cokoliv jiného. V oblasti rozpoznávání lidských akcí to může být pouhá silueta člověka nebo také komplexní model lidské kostry zachycující všechny klouby. Cílem procesu nazývaného jako extrakce příznaků je tedy redukce velkého množství informací dostupných ve videosekvencích do kompaktní, snadno zpracovatelné formy, přičemž v rámci redukce můžeme opominout pouze ty informace, které nejsou pro nás zajímavé a využitelné.

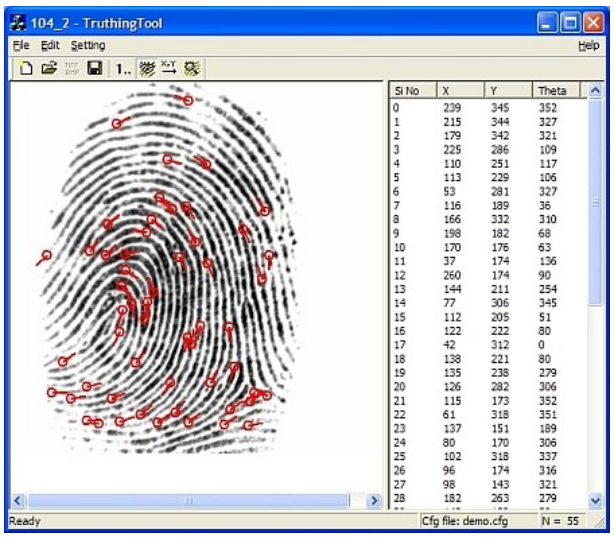
3.2 Prostorová reprezentace akcí

Prvním krokem tohoto druhu reprezentace akcí je extrakce příznaků z obrazu, přičemž se zabýváme hlavně postojem a pohyby člověka. Existuje však několik způsobů, jak reprezentovat lidské tělo. Hlavním rozdílem jednotlivých reprezentací je to, jak efektivní jsou v extrakci příznaků. Proto je můžeme rozdělit do tří hlavních kategorií:

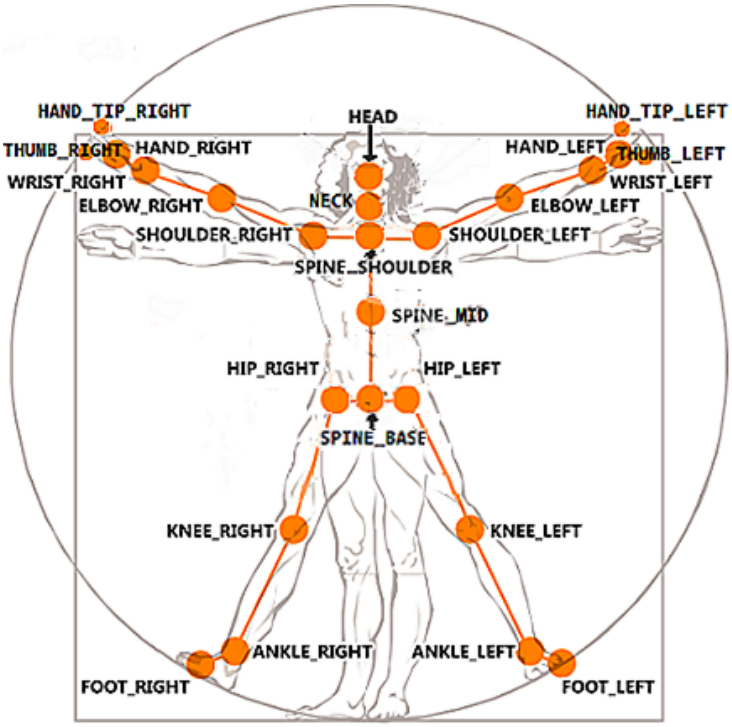
3.2.1 Metody založené na kostře lidského těla

Tyto metody reprezentují akce v prostoru vzhledem k tělu člověka. Z každého snímku videosekvence je vytvořena za pomocí různých dostupných příznaků lidská kostra. Na základě těchto modelů lidské kostry dochází poté k samotnému rozpoznávání akcí. Jedná se tedy o intuitivní a biologicky věrohodný přístup k rozpoznávání akcí.

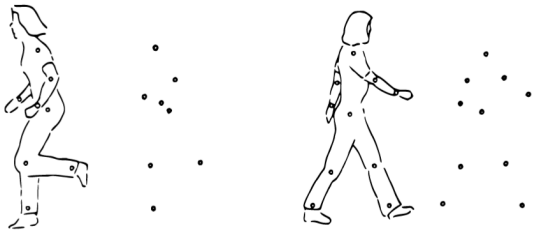
Vědcem Johanssonem bylo zjištěno, že člověk je schopen rozeznat prováděnou akci pouze z několika bodů lidského těla [6]. Například pokud jsou na vybraných kloubech člověka rozmístěny



Obrázek 6: Typickým příkladem procesu extrakce příznaků je například extrakce tzv. markantů (charakteristických znaků) z papilárních linií otisku prstu [7]



Obrázek 7: Rozmístění sledovaných bodů na těle člověka, pohledem knihovny Kinect for Microsoft SDK. [8]



Obrázek 8: Ilustrace schopnosti člověka rozeznat akci pouze z několika bodů umístěných na lidském těle. [6]

svítící body a sledovaná osoba je umístěna ve tmě, stačí toto člověku k rozpoznání akce (viz obrázek 8). Tento experiment byl proto počátkem diskuzí o tom, zda člověk rozpoznává akce přímo z 2D vzorů pohybu v obraze, nebo zda si nejdříve zrekonstruuje 3D scénu pohybu a až poté z něj akci rozpozná. Tyto dva přístupy nakonec vyústily k rozdělení strojového rozpoznávání akcí na dva oddíly:

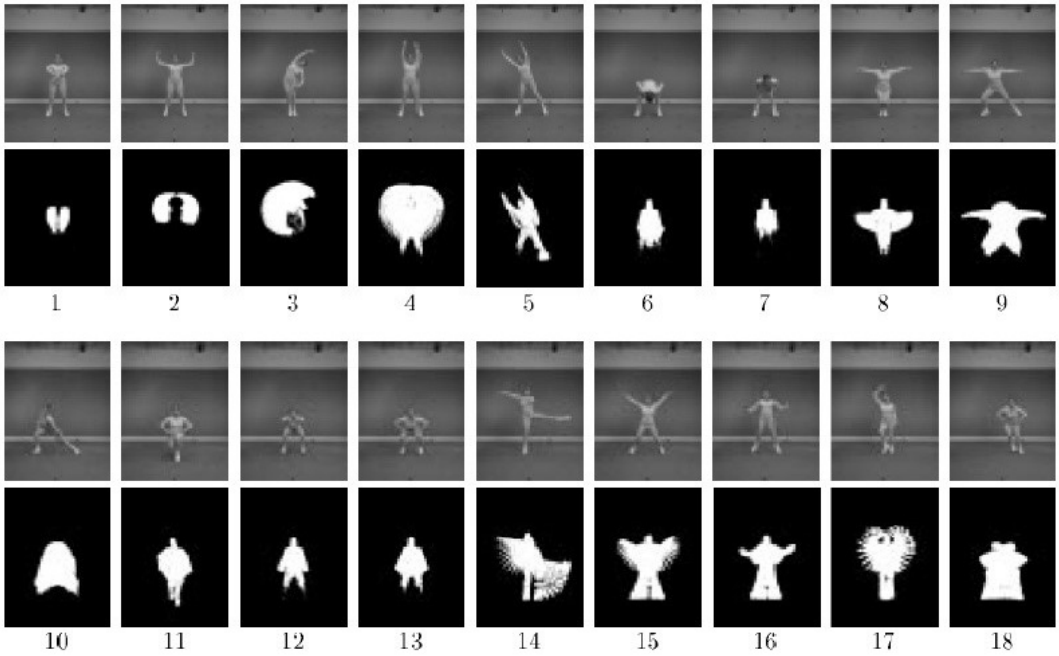
- rozpoznávání rekonstrukcí 3D kostry člověka,
- přímé rozpoznávání z 2D modelu kostry člověka.

Rozpoznávání pomocí rekonstrukce 3D kostry člověka se skládá ze dvou částí. První část se zabývá konstrukcí 3D kostry člověka, která je typicky reprezentována jako model složený z kloubů. Druhá část se zabývá samotným rozpoznáváním akcí, založeným na trajektoriích pohybu jednotlivých kloubů. Existují zde však dva problémy: vysoká variabilita pohybů člověka a jejich tvarů. Parametrický model lidského těla proto musí být vhodně zvolen. Oproti tomu přímé rozpoznávání nevyžaduje převod do 3D modelu a akce jsou tedy rozpoznávány přímo z 2D modelu kostry člověka.

3.2.2 Obrazové modely

Metody rozpoznávání obrazu založené na obrazových modelech nevyžadují, oproti metodám založených na kostře lidského těla, detekci a pojmenování jednotlivých částí těla člověka. Cílem je pouze detekovat oblast zájmu okolo sledované osoby, přičemž ve většině případů jsou pak příznaky vypočteny právě na základě tohoto ohraničení detekované oblasti. Ze zmíněných důvodů jsou obrazové modely mnohem jednodušší nežli modely založené na kostře člověka, a v důsledku toho mohou být prováděny mnohem efektivněji a robustněji. Paradoxně se však ukázalo, že jsou obě tyto metody schopny dosáhnout stejných výsledků s ohledem na mnoho kategorií pohybových akcí, avšak metody založené na obrazových modelech jsou velmi náchylné na změnu úhlů kamery vzhledem ke snímané osobě či velikosti samotného těla. Proto je nutno použít více vzorů, od různých osob, na základě kterých se poté bude určovat vykonávaná akce.

Jak již bylo zmíněno výše, jedním z důležitých kroků při použití metod založených na obrazových modelech je extrakce oblasti zájmu, neboli siluety člověka. Nejjednodušším případem

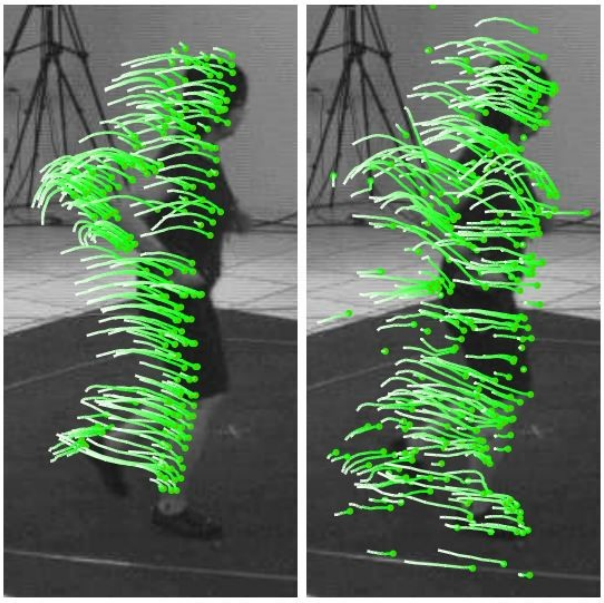


Obrázek 9: Obrazové modely různých pohybů. [9]

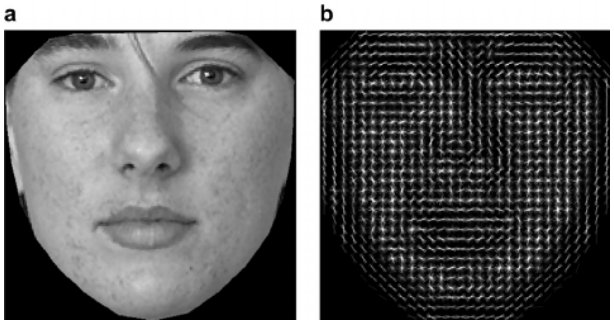
jsou sekvence, kdy se za snímaným člověkem nachází černé (případně jednobarevné) pozadí, a tak není nutno extrahovat příznaky, ale lze přímo porovnávat jednotlivé snímky sekvence s již existujícími vzory. V ostatních případech je nutno přistoupit k extrakci pozadí ze snímku, což může být, zejména u sekvencí s členitým pozadím, problém. Tyto problémy lze eliminovat použitím algoritmů známých jako Chamfer Distance nebo Shape Context a docílit tak siluety s jasnými okraji bez šumu a jiných nedokonalostí. Extrahované siluety jsou pak vhodným vstupem pro rozpoznávání akcí s tím, že jsou nenáchylné na změnu barvy, texturu nebo změny kontrastu snímku. Úspěšnost rozpoznávání však závisí ve velké míře na kvalitě a robustnosti algoritmu na odstranění pozadí snímku.

Další důležitou částí obrazových modelů je využití optického toku extrahovaného z po sobě následujících snímků. Reprezentace na základě toku obrazu nejsou závislé na odstranění pozadí snímku a mohou tak být lépe použitelné v mnoha případech. Tyto metody jsou založeny na předpokladu, že změny mezi jednotlivými snímky v sekvenci jsou iniciovány pohybem. Nereagují tak na změny textury, osvětlení, jasu a podobně.

Jednou z dalších neopomenutelných částí obrazových modelů je extrakce příznaků na základě gradientu. V oblasti počítačového vidění se gradientem chápe směr růstu intenzity nebo barvy v obraze. Při použití tohoto typu extrakce příznaků je každý snímek sekvence reprezentován histogramem těchto gradientů. Jedním z příkladů úspěšných metod extrakce příznaků na základě gradientu může být HOG deskriptor, který se osvědčil na detekci a rozpoznávaní akcí člověka. Tento deskriptor nepočítá histogram z celého snímku najednou, nýbrž si tento snímek rozdělí do



Obrázek 10: Znázornění optického toku při chůzi člověka. [10]

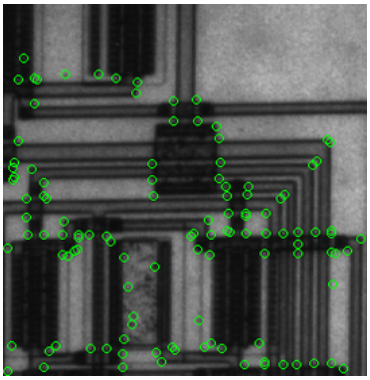


Obrázek 11: (a) původní snímek, (b) snímek po aplikaci HOG deskriptoru . [11]

několika vzájemně přesahujících se bloků, a následně spočítá histogram pro každý z těchto bloků separátně. Toto vede k vyšší přesnosti při extrakci příznaků. Příznaky na základě gradientů, stejně jako při využití optického toku, nejsou závislé na odstranění pozadí ze scény. Na rozdíl od optického toku jsou však gradienty schopny popsat obraz i v případě, kdy je scéna statická, což může mít své výhody i nevýhody (extrakce příznaků může být například ovlivněna objektem v pozadí).

3.2.3 Metody využívající obrazové statistiky

Metody využívající obrazové statistiky jsou založeny na dekompozici snímku nebo celé videosekvence do menších oblastí, které nejsou závislé na částech lidského těla nebo souřadnicovém systému. Z těchto oblastí jsou poté získávány statistiky lokálních příznaků, na základě kterých poté dochází k samotnému rozpoznávání akcí. Výhodou metod využívající obrazové statistiky je tedy nezávislost na pojmenování částí lidského těla, jeho detekci a lokalizaci. Tento přístup



Obrázek 12: Příklad nalezení lokálních příznaků pomocí detekce rohů v obraze. [13]

je typicky založen na strategii shora dolů, v rámci které jsou nejprve detekovány zájmové body v obraze (zejména v strukturách jako jsou rohy nebo v místech, kde je náhlá změna v jase či barvě vzhledem k okolí) a poté je každé oblasti v obraze přiřazena množina předpřipravených příznaků.

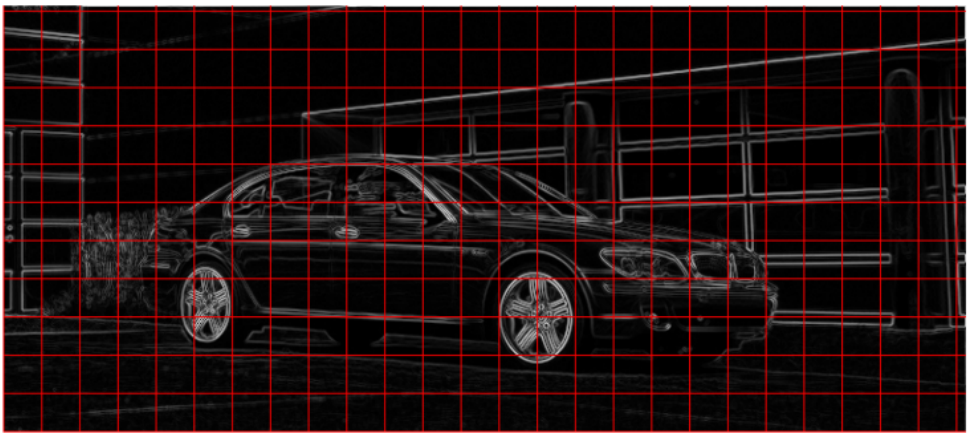
3.3 Příznaky

K strojovému popisu jakékoliv akce vykonávané člověkem jsou zapotřebí náležité vstupy. V případě rozpoznávání akcí je cílem získat unikátní vlastnosti akce, které budou s jinou akcí nezaměnitelné. Tyto vlastnosti, nazývané také jako příznaky, následně umožní jejich přesnou klasifikaci. Ideální příznak by měl splňovat tyto požadavky [12]:

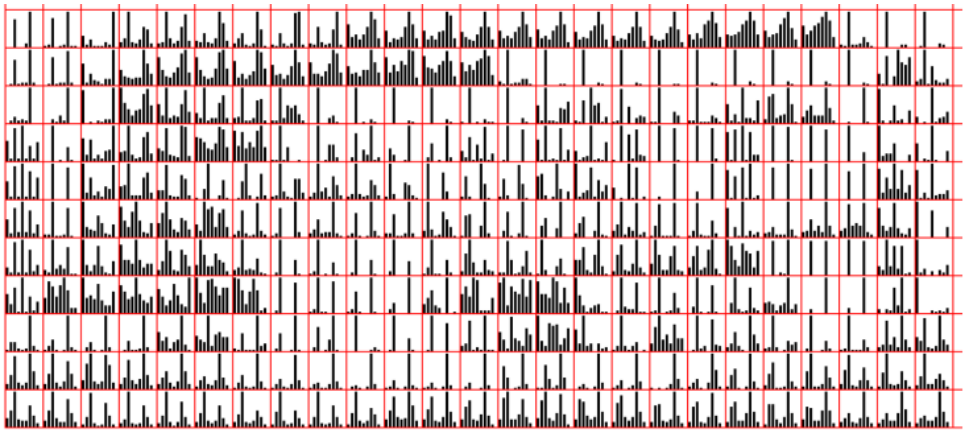
- Invariantnost – nezávislost na změně měřítka, kontrastu ve scéně, jasu, rotaci, translaci,
- Spolehlivost – pokud jsou akce ve stejné třídě, musí mít jejich příznaky podobné hodnoty,
- Diskriminabilita – příznaky akci pro různé třídy musí být rozlišitelné,
- Efektivita výpočtu – příznak musí být jednoduše, rychle a správně detekovatelný,
- Časová invariance – příznak se nemění při použití dynamických obrazů (videosekvencí).

3.3.1 HoG

HoG (Histogram of oriented gradients) je jeden z úspěšných deskriptorů založených na extrakci příznaků na základě gradientu [14]. Základní myšlenkou této metody je skutečnost, že vzhled a tvar nějakého objektu může být také dobře charakterizován distribucí lokálních gradientů nebo směrem hran v obraze. V praxi to znamená, že je snímek rozdělen do malých částí, nazývaných jako buňky, přičemž pro každou takovou buňku je na základě jednotlivých pixelů vypočítán lokální histogram gradientů, či orientace hran. Následná kombinace všech vzniklých histogramů je



Obrázek 13: Snímek rozdělený do buněk. [15]

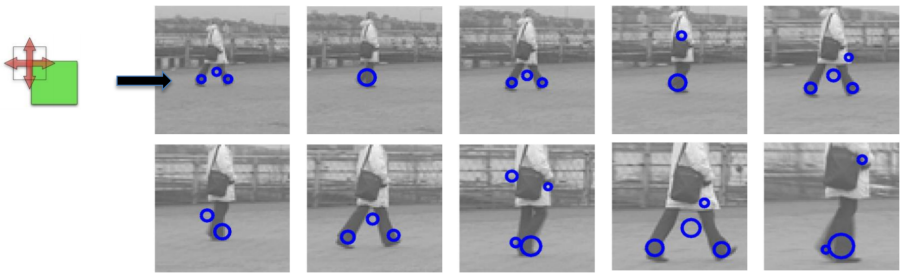


Obrázek 14: Vyobrazení lokálních histogramů pro jednotlivé buňky. [15]

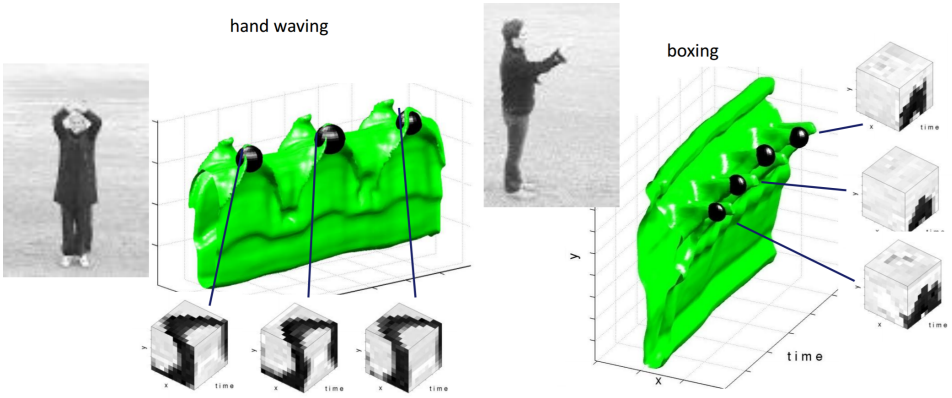
výslednou reprezentací tohoto deskriptoru. Pro zvýšení úspěšnosti a výkonnosti tohoto deskriptoru je vhodné provést normalizaci kontrastu jednotlivých lokálních histogramů. Toto může být provedeno pomocí výpočtu měřítka intenzity nad větší oblastí snímku, nazývanou jako blok. Jednotlivé bloky čtvercového nebo kruhového tvaru se obvykle překrývají, což znamená, že každá buňka snímku je zapojena do výsledného deskriptoru vícekrát. Poté za využití hodnoty získané z výpočtu měřítka intenzity dochází k samotné normalizaci všech buněk obsažených v daném bloku. Výsledkem této normalizace je menší náchylnost k ovlivnění deskriptoru změnami v osvětlení či stíny v obraze.

3.3.2 STIP – Space-Time Interest Point

STIP (Space-Time Interest Point) je metoda výběru příznaků založených na detekci zájmových bodů v obraze, zejména rohů a jejich následné sledování v čase. Tato metoda byla navržena vědcem I. Laptevem v roce 2005 [16] a je rozšířením 2D Harrisova detektoru rohů. Tyto zájmové



Obrázek 15: Vyobrazení STIP při chůzi člověka. [17]



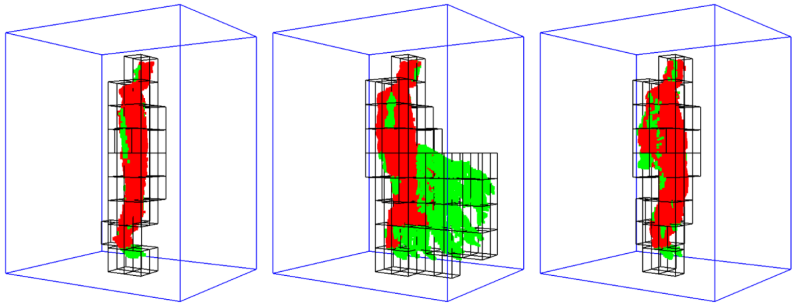
Obrázek 16: Vyobrazení STIP v časoprostoru. [17]

body v obraze jsou atraktivní díky jejich vysoké vypovídací hodnotě, přičemž jsou stabilní i při změně úhlu pohledu na sledovaný objekt či osobu a mohou být definovány následovně:

- Mají jasnou, matematickou definici.
- Mají jasně definovanou pozici v obraze.
- Struktura obrazu v okolí zájmového bodu má svou vypovídající hodnotu.
- Jsou stabilní vůči lokálním i globálním odchylkám v obraze (změny jasu či osvětlení).
- Mohou být spolehlivě vypočteny s vysokou mírou opakovatelnosti.
- Mohou být vypočteny nezávisle na měřítku.

3.3.3 STOP – Space-Time Occupancy Pattern

Příznaky Space-Time Occupancy Pattern byly vyvinuty k rozpoznávání akcí v 3D prostoru. Tyto příznaky využívají jako vstup přímo hloubkové mapy bez dodatečné extrakce pozic kloubů v prostoru, jelikož tato extrakce nemusí být vždy zcela přesná [18]. Osy prostoru i času sekvence



Obrázek 17: Vyobrazení STOP v časoprostoru. [18]

jsou rozděleny do několika segmentů. Pro každou sekvenci hloubkových dat je tedy definována mřížka v časoprostoru. Výhodou příznaků STOP je fakt, že zachovávají jak prostorovou, tak i časovou informaci mezi jednotlivými buňkami v časoprostoru, které jsou definovány již zmiňovanou mřížkou, přičemž jsou stále dostatečně flexibilní k uchování informace o akcích uvnitř buněk. Informace uvnitř buněk se typicky skládá ze siluety člověka či pohybujících se částí těla, a tak jsou tyto informace důležité k rozpoznávání akcí. Vypovídající hodnotou a zároveň výstupním příznakem je tedy míra obsazenosti buněk v prostoru.

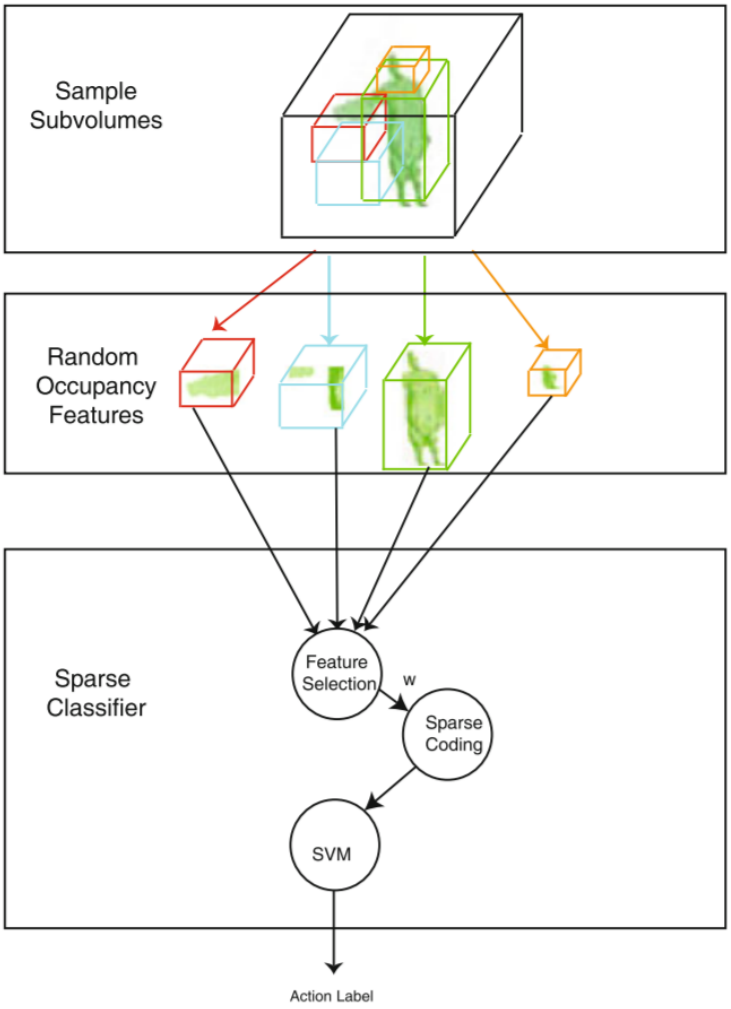
3.3.4 ROP – Random Occupancy Patterns

Příznaky označované jako Random Occupancy Patterns (ROP) byly vyvinuty zejména na základě nedostatečné spolehlivosti algoritmů založených na kostře lidského těla v případě, kdy v obraze nastane několikanásobná okluze [19]. Okluzí je myšlen stav, kdy nejsou vidět všechny části obrazu, které jsou nutné k úspěšné konstrukci modelu kostry člověka, například překrytí některých kloubů jinou částí těla. ROP proto využívá jako vstup přímo hloubkové mapy bez dodatečné konstrukce lidské kostry.

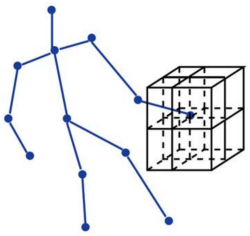
Při použití těchto příznaků je s trojrozměrnou scénou akce zacházeno jako s 4D objektem, přičemž jednotlivé ROP příznaky jsou poté extrahovány z náhodně získaných podoblastí různých velikostí a v různých lokacích celé scény. Body v jednotlivých podoblastech jsou sečteny za použití sigmoidální funkce. ROP příznaky jsou tedy poměrně velkými oblastmi scény, což zapříčiňuje odolnost vůči šumu a zároveň jsou méně citlivé na okluzi, protože získávají informace pouze z těch oblastí, které mají největší popisující hodnotu pro danou akci. Dále jsou za použití regularizační metody Elastic-Net vybrány nejužitečnější příznaky.

3.3.5 LOP - Local Occupancy Pattern

Použití pozic lidských kloubů v 3D prostoru nemusí být vždy zcela dostatečné k reprezentaci akcí. Je tomu tak zejména v případech, kdy akce zahrnuje interakci člověka s jiným objektem. Příkladem této interakce může být oblékání trika, kdy je při uchopení trika zakryta ruka a následně při samotném oblékání trika i hlava a další částí těla člověka. Tato informace je poté



Obrázek 18: ROP - Průběh metody rozpoznávání akcí popisované v [19]. Svrchu: příklad vybraných podoblastí, ROP příznaky, klasifikace akcí (selekce příznaků, sparse coding, SVM)



Obrázek 19: LOP - Vyobrazení vzoru obsazenosti okolo zápěstí člověka. [19]

využita k rozlišení oblékání trika od jiných akcí. Interakce mezi člověkem a jiným předmětem je charakterizována právě jako Local Occupancy Pattern (LOP) – vzor obsazenosti v daném bodě [19]. Je tedy důležité navrhnout takový příznak, který dokáže spolehlivě popsat hloubku v prostoru pro jednotlivé klouby.

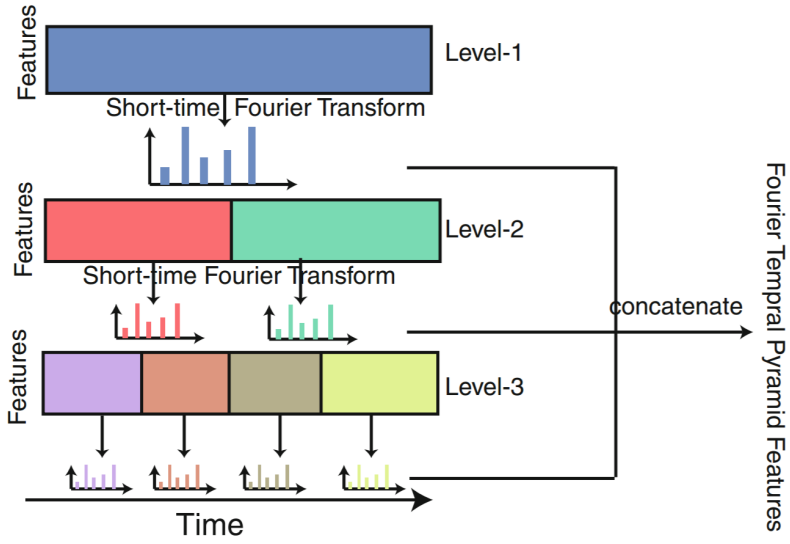
V rámci získání LOP příznaků se vychází z toho, že pro každý snímek sekvence existuje point cloud (množina bodů v souřadnicovém systému), který je vygenerován ze vstupních hloubkových map. V okolí každého z kloubů na lidském těle je vytvořena mřížka, která obsahuje tzv. biny, což jsou buňky dané mřížky. Následně jsou sečteny body z point cloudu, které spadají do příslušných binů mřížky a je aplikována sigmoidální funkce k získání příznaků pro jednotlivé biny. LOP příznak jednotlivých kloubů je tedy vektor skládající se z příznaků všech binů v mřížce okolo daného kloubu.

3.3.6 FTP – Fourier Temporal Pyramid

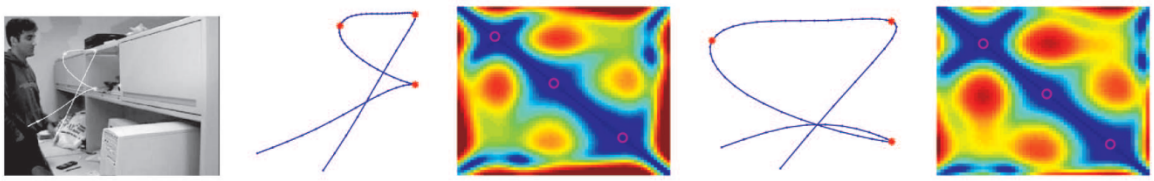
Příznak s názvem Fourier Temporal Pyramid (FTP) [19] byl vyvinut hlavně z důvodu nepřesnosti levných hloubkových kamer, a to jak prostorové, tak i časové. Dalším úkolem při návrhu FTP bylo vyvinout takový příznak, který je schopný účinně popsat časovou strukturu akce. Jednotlivé akce totiž mohou být složeny z více podakcí, například zvednutí předmětu ze země – nejdříve se člověk ohne k zemi a poté uchopí a zvedne předmět. Aby mohla být zachycena časová struktura akce, musí být krom výpočtu globálního Fourierova koeficientu také celá akce rekurzivně rozdělena do pyramidy, kde je následně na všechny její části aplikována Fourierova transformace, viz obrázek 20. Finální vektor je tedy zřetězení Fourierových koeficientů všech segmentů pyramidy.

3.3.7 Temporal Self-Similarities

Tato metoda rozpoznávání akcí využívá matice soběpodobnosti, což je grafické vyjádření podobných sekvencí v sérii dat. Podobnost může být vyjádřena různě, například pomocí prostorové vzdálenosti, korelace nebo porovnáním lokálních histogramů. V případě metody Temporal Self-Similarities [20] se jedná o prostorovou vzdálenost mezi jednotlivými klouby na těle člověka. Diagonála matice odpovídá porovnání snímku se sebou samým, což znamená že zde dochází k stoprocentní podobnosti, a proto jsou na ní zobrazovány nuly.



Obrázek 20: FTP - Ukázka rozložení akce do podakcí směrem shora dolů. [19]



Obrázek 21: Na obrázcích (b) a (d) lze vidět trajektorie pohybu ruky při otevírání skříňky, přičemž každá z trajektorií patří jinému člověku. Ke každé této trajektorii náleží vypočítaná matice soběpodobnosti (c) a (e). Na obou maticích soběpodobnosti můžeme pozorovat podobné vzory. [20]

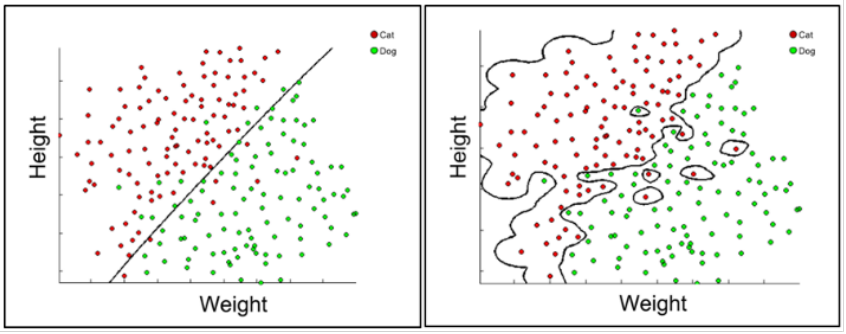
Bylo by velmi neefektivní a výkonově náročné počítat matici soběpodobnosti pro každý kloub lidského těla jednotlivě. V každém snímku sekvence tedy dochází k zprůměrování hodnot Euklidovské vzdálenosti (tedy vzdálenosti v prostoru) všech dvojic kloubů. Výstupem této metody je tedy čtvercová matice, která je následně použita jako vstup do klasifikátoru akcí.

3.4 Učení a klasifikace akcí

Algoritmy klasifikace akcí v dnešní době stále potřebují jakousi předlohu dané akce, na základě které pak dochází k samotné klasifikaci neboli porovnávání předlohy s právě prováděnou akcí. Kvůli odlišnosti stavby postavy jednotlivých osob, jejich oblečení, pozadí akce ve videosekvenci a podobně (ostatně jak už je zmíněno i v předchozích kapitolách), musí být použito hned několik vzorů od různých osob pro jednu akci. Existují však i algoritmy, které nevyžadují předchozí kategorizaci akcí.

Tabulka 1: Typické vzory vyskytující se v maticích podobností a jejich význam

Vzor	Význam
Homogennost, stejnorodá plocha	Neprobíhá žádná akce
Slábnutí v rozích	Nestacionární data
Opakující se vzory	Opakující se proces
Jednotlivé osamocené body	Silné výkyvy v procesu
Diagonální linie (paralelní k hl. diagonále)	Trajektorie pohybu prochází stejnou oblastí v rozdílnou dobu
Diagonální linie (kolmé k hl. diagonále)	Trajektorie pohybu prochází stejnou oblastí v rozdílnou dobu, ale v opačném čase
Dlouhé zahnuté liniové struktury	Trajektorie pohybu prochází stejnou oblastí v rozdílnou dobu, ale jinou rychlostí



Obrázek 22: Na levém grafu lze vidět správně naučenou neuronovou síť. Na pravém grafu došlo k přeučení neuronové sítě. [21]

3.4.1 Učení s učitelem

V oblasti rozpoznávání akcí je učení s učitelem daleko rozšířenější formou, nežli učení bez učitele [22]. Je tomu tak z důvodu, že cílem procesu rozpoznávání akcí je zpravidla naučit stroj rozpoznávat právě ty akce, které požadujeme. Obecně vzato, učení s učitelem je proces, kdy jsou předem definovány a popsány jednotlivé akce, které jsou poté porovnávány s akcemi právě vykonávanými.

Učení s učitelem je jedna z hlavních metod pro zdokonalování neuronových sítí a rozhodovacích stromů. Obě tyto techniky jsou velmi závislé na předchozím učení. V případě neuronových sítí je cílem učení určit a následně minimalizovat odchylku sítě. V případě rozhodovacích stromů je cílem určit, který atribut poskytne nejvíce informací k zařazení dané akce. Jednou z nejdůležitějších věcí v rámci procesu učení je volba správné množiny trénovacích dat tak, aby nedošlo k přílišnému přizpůsobení rozhodovacích kritérií vzhledem k této množině. Tento stav se nazývá jako přeučení a může k němu dojít například při volbě příliš velké množiny trénovacích dat nebo pokud je systém příliš složitý.

3.4.2 Učení bez učitele

V oblasti rozpoznávání akcí je cílem učení bez učitele naučit stroj rozpoznávat akce, aniž by měl jakýkoliv vzor této akce [22]. V současné době existují dva přístupy k učení bez učitele. První přístup je založen na tom, že v rámci procesu učení nedochází k přímé kategorizaci akcí, ale pouze k potvrzení, zda systém při klasifikaci uspěl či nikoliv. Za úspěšnou klasifikaci získá systém body, za neúspěšnou je naopak ztrácí. Cílem učení tedy není dosáhnout správné klasifikace, nýbrž získat co nejvíce bodů. Druhý typ učení bez učitele je nazýván jako shlukování. Cílem tohoto typu učení není maximalizovat účinnost rozhodovací funkce, ale nalézat podobnosti v trénovacích datech. Předpokládá se, že získané shluky trénovacích dat reprezentují jednotlivé akce. Také u učení bez učitele, stejně jako u učení s učitelem, může dojít k přeučení.

3.4.3 SVM - Support Vector Machine

Support Vector Machine je jedna z metod strojového učení s učitelem určena zejména ke klasifikaci a regresní analýze [23]. Vstupní data do této metody strojového učení jsou rozdělena do dvou sad, a to trénovací a testovací. Trénovací sada obsahuje data ve formě vektorů v n-dimenzích. Ke každému trénovacímu vektoru také náleží informace, do které třídy dat patří. Trénovací sada musí být zvolena s ohledem na co možná nejlepší rozlišitelnost jednotlivých tříd v testovací sadě, protože právě na základě trénovací sady jsou zvoleny parametry klasifikační funkce SVM. Nejjednodušším způsobem rozdělení dat dvou skupin je použití lineárního klasifikátoru. Ne vždy lze však vstupní data lineárně rozdělit, a proto musí být přistoupeno k tzv. jádrovému triku, kdy jsou data transformována z původního prostoru příznaků do prostoru vyšší dimenze. V tomto prostoru vyšší dimenze jsou data lineárně separabilní.

4 Implementace vybrané metody

Pro implementaci vybrané metody rozpoznávání akcí byl vybrán programovací jazyk C++ [24] z důvodu jeho rozšířenosti, využití knihovny OpenCV, která podporuje tento jazyk, a jeho multiplatformnosti.

4.1 Microsoft Kinect

Microsoft Kinect je RGB-D senzor poskytující jak klasický barevný obraz, tak i výstup v podobě hloubkových map [25, 26]. Zařízení bylo původně vyvinuto pro herní konzole XBOX k ovládání her tělem bez jakéhokoliv ovladače. Kinect je ovšem také, zejména kvůli své hloubkové kameře, využitelný pro počítačové vidění. Tato kamera byla zvolena pro zachycení vstupních dat z důvodu její dostupnosti, rozšířenosti, ceny a kvality. Kinect obsahuje dva senzory:

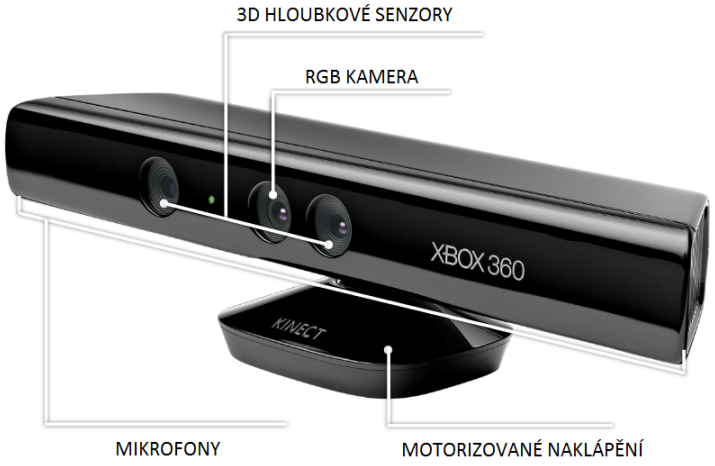
- RGB kamera – Klasická kamera poskytující tři barevné složky obrazu. Poskytuje rozlišení 640 x 480 pixelů při snímkovací frekvenci 30 Hz nebo rozlišení 1280 x 1024 pixelů při snímkovací frekvenci 10 Hz.
- 3D hloubková kamera – Hloubkové mapy jsou získávány na základě analýzy obrazu pomocí strukturovaného světla. Do scény je vyslán předem známý infračervený signál ve formě bodového záření. Hloubka je následně určena na základě jeho deformace o objekty ve scéně. Při praktickém využití musí být vzdálenost objektu od senzoru 0,8 – 3,5 m. Výstupní video má rozlišení 640 x 480 při snímkovací frekvenci 30 Hz.

4.2 Dataset

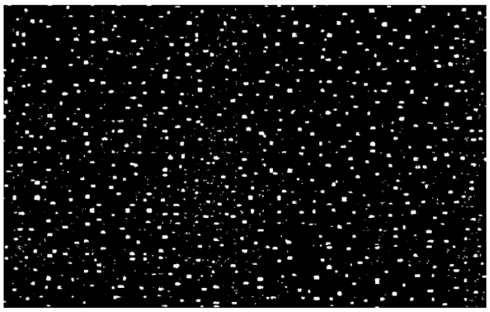
Jako vstupní informace k dalšímu zpracování byl využit dataset UTKinect-Action3D [27]. Tento dataset byl zachycen právě zmiňovaným zařízením Microsoft Kinect a je dostupný k jakémukoliv užití zdarma. Dataset se skládá ze čtyř částí:

- RGB snímky v rozlišení 480x640 a formátu JPG.
- Hloubkové mapy v rozlišení 320x240 a formátu XML, mapy jsou uloženy pomocí knihovny OpenCV.
- Textové soubory s pozicemi jednotlivých kloubů.
- Popis jednotlivých sekvencí ve formě textového souboru.

Pro tuto práci byly jako vstupní informace využity textové soubory s pozicemi jednotlivých kloubů. Tyto pozice byly získány pomocí 3D hloubkové kamery a následně převedeny do textového souboru za použití knihovny Kinect for Microsoft SDK. Dataset obsahuje deset takovýchto



Obrázek 23: Zařízení Kinect. [25]]



Obrázek 24: Vzor infračerveného signálu vysílaného kamerou Kinect. [25]

textových souborů, přičemž každý pochází od jiného člověka, což zajišťuje jistou míru různorodosti prováděných akcí. Jednotlivé soubory dále obsahují deset různých akcí (chůze, usednutí, postavení se, zvednutí předmětu, nesení předmětu, hod, strčení, táhnutí, mávání, tleskání) a jejich formát je následující: každý řádek souboru obsahuje data týkající se pouze jednoho snímku z kamery, přičemž první číslo na řádku označuje číslo snímku a následující čísla jsou vždy 3D pozice (x, y, z) jednotlivých kloubů či míst na těle člověka. Souřadnice x, y a z jsou vzdálenosti v metrech vzhledem ke středu snímače kamery. Pro účely této práce byly jednotlivé soubory dále rozděleny podle prováděných akcí.

4.3 OpenCV

OpenCV (Open Source Computer Vision Library) [28] je multiplatformní open-source knihovna obsahující stovky algoritmů pro manipulaci s obrazem vydávána pod licencí BSD, což znamená, že je zdarma pro akademické, ale i komerční využití. Knihovna je vytvořena pro použití s programovacími jazyky C, C++, Python a Java, nyní je ale také použitelná na systémech Android či iOS. Knihovna obsahuje následující moduly:

- Core functionality - kompaktní modul definující základní datové struktury, zahrnující vícerozměrné pole Mat a další základní funkce využívané ostatními moduly.
- Image processing - modul obsahuje lineární i nelineární obrazové filtry, geometrické transformace obrazu (změna velikosti, změna perspektivy apod), konverze barevných prostorů, histogramy.
- Video - modul pro analýzu videa obsahuje algoritmy pro detekci pohybu, odstranění pozadí, sledování pohybu objektu.
- Calib3d - modul obsahující základní metody pro práci s 3D videem.
- Features2d - modul obsahující detektory příznaků a deskriptory.
- Objdetect - detekce objektů předdefinovaných tříd (např. obličej, uši, oči, lidé, auta apod).
- Highgui - jednoduché metody pro práci s uživatelským rozhraním.
- Video I/O - rozhraní pro zachytávání videa a kodeky videí.
- Gpu - modul obsahující algoritmy akcelerované pomocí grafické karty počítače.

4.4 Implementace

Implementovaný algoritmus rozpoznávání akcí je založen na kostře lidského těla, potažmo pozicích jednotlivých kloubů v 3D prostoru. O extrakci pozic kloubů či jiných významných bodů na těle z hloubkových map, poskytnutých kamerou Kinect, se stará knihovna s názvem Kinect for

```
1638  -0.293143 -0.053226 2.178769 -0.298684 0.011930 2.174270 -0.325240 0.337562
2.164468 -0.316811 0.511784 2.124844 -0.456123 0.238614 2.157190 -0.498608 -0.014710
2.181057 -0.492053 -0.207138 2.151639 -0.478908 -0.296407 2.145413 -0.177002
0.248871 2.150819 -0.066447 0.242802 1.849756 0.019716 0.227045 1.692854 0.056137
0.217755 1.625520 -0.352951 -0.118946 2.176429 -0.340258 -0.555926 2.261564 -
0.348894 -0.875948 2.323337 -0.344711 -0.920092 2.255725 -0.221754 -0.113436
2.180135 -0.147428 -0.565379 2.243524 -0.104253 -0.901027 2.331328 -0.107219 -
0.944849 2.256757

1640  -0.285061 -0.052790 2.174124 -0.290119 0.012422 2.170186 -0.313876 0.338274
2.162967 -0.308318 0.512925 2.123237 -0.447989 0.241714 2.160347 -0.486164 -0.013906
2.183495 -0.480905 -0.206445 2.154629 -0.469840 -0.295048 2.148682 -0.166819
0.247724 2.149058 -0.061235 0.234501 1.858739 0.032974 0.204818 1.695222 0.049011
0.236714 1.674064 -0.344947 -0.118526 2.173190 -0.340715 -0.553546 2.257448 -
0.346342 -0.875805 2.324065 -0.344830 -0.918299 2.248892 -0.214462 -0.113258
2.173637 -0.142381 -0.565747 2.248824 -0.102223 -0.900671 2.333460 -0.110096 -
0.943870 2.257681
```

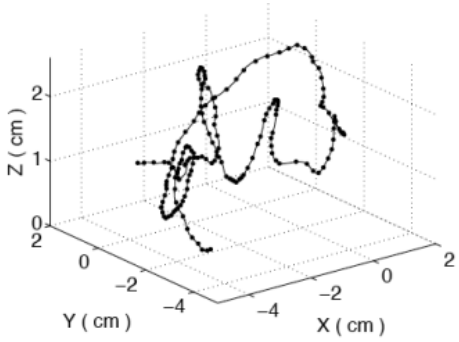
Obrázek 25: Struktura vstupního souboru obsahujícího pozice kloubů na těle člověka. Žlutě označený text reprezentuje číslo zachyceného snímku, další čísla jsou již samotné pozice kloubů či jiných významných míst na těle. Zeleně označená je pozice středu pánve, světle modře střed páteře, tmavě modře střed mezi rameny, fialově hlava, červeně levé rameno, šedě levý loket atd.

Microsoft SDK, která obsahuje patřičné metody. Tyto metody jsou v současné době aktuální a dostatečně výkonné [29].

Akce jsou tedy po aplikaci zmiňovaných metod reprezentovány množinou bodů v trojrozměrném prostoru a jsou uloženy v textovém souboru. Pro další využití je tedy nutné tyto soubory načíst do programu. O toto se v případě této implementace stará třída Action, potažmo její metoda $readFromFile(StringfileName)$, kdy vstupem je právě zmiňovaný textový soubor. Tato metoda načte data do předem připravených datových struktur. Pro každý kloub či jiné významné místo na těle člověka je připraven $Vector$. Po načtení dat je nutno začít pracovat s reprezentací akce. Po spojení těchto 3D bodů přímkami je získána trajektorie dané akce.

Proto je pro zjednodušení a vyhlazení trajektorie akce využit následující postup: každá trajektorie je rozdělena na určitý počet sekcí, přičemž každou z těchto sekcí je dále proložena přímka pomocí funkce $fitLine$ z knihovny OpenCV, která přijímá následující parametry:

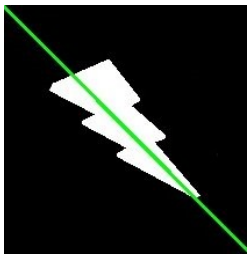
- *points* – Vstupní vektor 2D nebo 3D bodů uložených ve $std :: vector <>$ nebo Mat.
- *line* – Výstupní parametr obsahující výstupní přímku. V případě vstupních 2D bodů se jedná o vektor o čtyřech prvcích (například $Vec4f$), - $(vx, vy, x0, y0)$, kde (vx, vy) je normálový vektor a $(x0, y0)$ je bod na přímce. V případě vstupních 3D bodů se jedná o vektor



Obrázek 26: Vyobrazení trajektorie jednoho kloubu v prostoru [30]

```
fitLine(groupOfPoints, tempDirection, CV_DIST_L2, 0, 0.01, 0.01);
```

Obrázek 27: Příklad využití funkce *fitLine*.



Obrázek 28: Příklad využití funkce *fitLine*. [28]

o šesti prvcích (například $Vec6f$) - $(vx, vy, vz, x0, y0, z0)$, kde (vx, vy, vz) je normálový vektor a $(x0, y0, z0)$ je bod na přímce.

- *distType* – Vzdálenost využitá minimalizační metodou M-odhad.
- *param* – Numerický parametr pro různé typy vzdáleností. V případě použití 0 je automaticky zvolena optimální hodnota tohoto parametru.
- *reps* – přesnost rádiusu (vzdálenost mezi originálním bodem a vytvářenou přímkou), osvědčená hodnota, která je také doporučována v oficiální dokumentaci knihovny OpenCV je 0,01.
- *aeps* – přesnost úhlu, osvědčená hodnota, která je také doporučována v oficiální dokumentaci knihovny OpenCV je 0,01.

Po aplikování funkce $fitLine$ na jednotlivé části trajektorie pohybu je získáno několik vektorů v 3D prostoru. Pro účely rozpoznávání akcí jsou důležité pouze normálové vektor, tedy směry, kam přímka míří. Původní akce, která se odehrává v 50 snímcích byla popsána 50 body v trojrozměrném prostoru, celkem tedy 150 hodnotami. Nyní, po rozdělení její trajektorie na N částí, například 5, je akce popsána 5 body v trojrozměrném prostoru, tedy 15 hodnotami. Tento počet hodnot je již vhodným vstupem do SVM klasifikátoru.

Pro klasifikaci akcí byla využita třída SVM z knihovny OpenCV, což je klasifikátor založený na LibSVM [31]. Než se začne využívat třída SVM pro klasifikaci, je nutno její instanci nejdříve nastavit, k čemuž existují základní tři parametry: $setType$, $setKernel$ a $term\_crit$. Pomocí parametru $term\_crit$ se určují kritéria pro ukončení iterativního výcvikového procesu SVM. Lze určit toleranci a maximální počet iterací. Pomocí parametru $setType$ se nastavuje typ formulace úlohy:

- C_SVC – C-SVM klasifikace, kde C je regularizační parametr používaný k aplikaci penalizace pro body, které nebyly správně určeny a napomáhá tak k zpřesňování klasifikace. C nabývá hodnot od 0 do nekonečna.
- NU_SVC – Nu-SVM klasifikace, kde nu je regularizační parametr používaný k aplikaci penalizace pro body, které nebyly správně určeny, a napomáhá tak k zpřesňování klasifikace. Nu nabývá hodnot od 0 do 1.
- ONE_CLASS – klasifikace, při níž jsou všechna trénovací data ve stejné třídě, SVM vytvoří hranici kolem těchto trénovacích dat a oddělí je tak od ostatních (neznámých) tříd.
- EPS_SVR a NU_SVR – typy užívané pro regresní analýzu.

Pomocí parametru *setKernel* se určuje typ použité SVM kernel funkce:

- *LINEAR* - tento kernel je vhodný pro lineárně oddělitelná data. Data jsou zpracovávána v původním n-dimensionálním prostoru.
- *POLY* - tento kernel je vhodný pro data, která jsou v původním prostoru oddělitelná křivkou. Data se převádí do prostoru o vyšší dimenzi, kde jsou již lineárně oddělitelná.
- *RBF* - tento kernel je vhodný pro data, která jsou v původním prostoru oddělitelná kružnicí, či elipsou. Data se převádí do prostoru o vyšší dimenzi, kde jsou již lineárně oddělitelná.

Pro samotné trénování SVM se využívá metody *train*, která přijímá jako parametry matici s trénovacími daty, popisky jednotlivých akcí a informaci, zda každý trénovací vzorek odpovídá řádku v trénovací matici.

Dalším krokem po trénování SVM je samotné určování tříd testovacích dat. Toto se děje pomocí metody *predict*, která přijímá jako parametr matici, kde každý řádek matice odpovídá jednomu vzorku testovacích dat. Výstupem této metody je číslo znázorňující třídu testovaného vzorku.

4.5 Testování

Testování je důležitou součástí vývoje softwaru. U implementovaného algoritmu bude důraz kladen na testování z hlediska přesnosti určení prováděné akce. Testování bude prováděno na běžně dostupném laptopu následující konfigurace:

- CPU: Intel Core i5-6200U 2.30GHz
- GPU: Intel HD Graphics 520
- RAM: 8GB
- SSD: Samsung SSD PM851 256 GB, čtení: 467 MB/s, zápis: 265 MB/s
- OS: Windows 10 Home

Pro účely testování byly z datasetu vybrány následující tři akce: mávání rukou, tleskání, předpažení a následné přitáhnutí předmětu. Tyto akce byly prováděny pouze pažemi člověka, a proto nebyly pro výpočet příznaků použity všechny klouby na těle, ale jenom ty, které se do dané akce zapojují nejvíce. V případě této implementace se jedná o zápěstí levé ruky, loket levé ruky, levé rameno, zápěstí pravé ruky, loket pravé ruky a pravé rameno. Testovací sada se vždy skládala z 10 provedení akce všech tříd, přičemž každá akce byla prováděna jiným subjektem. Celá sada byla rozdělena na trénovací data a testovací data. Pro trénování SVM bylo použito vždy n-subjektů, které prováděly akce pro všechny třídy, přičem zbývající subjekty z datasetu byly použity jako testovací data. Například v případě použití 7 subjektů jako trénovacích byly použity 3 subjekty jako testovací. Trénovací sadu tvořilo maximálně 6 subjektů.

Tabulka 2: Míra úspěšnosti při rozdělení trajektorie na 5 částí a užití lineárního kernelu SVM

Počet natrénovaných subjektů	Úspěšnost [%]
1	42,86
2	22,23
3	72,73
4	83,34
5	54,55
6	25

Tabulka 3: Míra úspěšnosti při rozdělení trajektorie na 5 částí a užití polynomického kernelu SVM

Počet natrénovaných subjektů	Úspěšnost [%]
1	42,86
2	20
3	50
4	71,43
5	40
6	28,57

Míra úspěšnosti rozpoznávání byla závislá zejména na nastavení SVM klasifikátoru, volbě trénovací sady a počtu částí, na které je rozdělena trajektorie dané akce. Míra úspěšnosti byla vypočítána za využití metody F1 score [32] následujícím vzorcem:

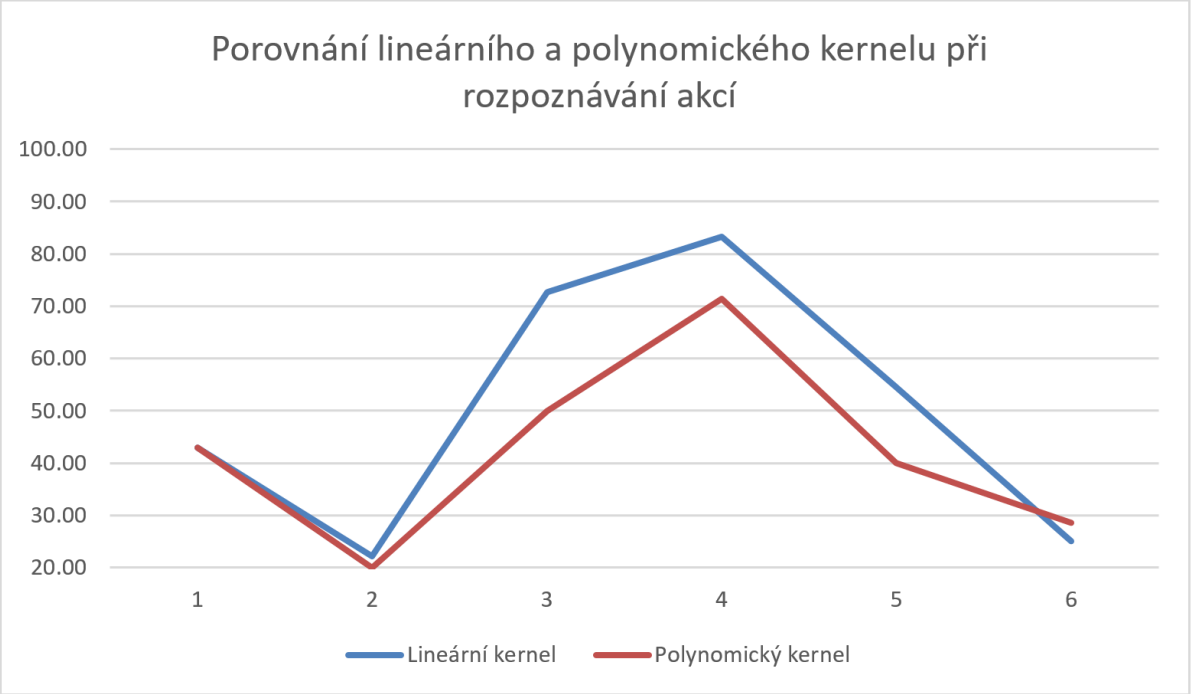
$$F_1 = \frac{2TP}{2TP + FP + FN}$$

kde TP jsou skutečně pozitivní předpovědi, FP falešně pozitivní a FN falešně negativní.

Při rozdělení trajektorie na 5 částí a užití lineárního kernelu SVM bylo nejlepších výsledků dosaženo použitím 4 trénovacích subjektů, kdy úspěšnost dosahovala 83,34%. S přibývajícím počtem trénovacích subjektů začalo docházet k tzv. přetrénování SVM (rozhodovací funkce je příliš přizpůsobena množině trénovacích dat) a úspěšnost určení akce se snižovala. V tabulce 2 lze pozorovat míru úspěšnosti vzhledem k počtu natrénovaných subjektů.

Při rozdělení trajektorie na 5 částí a užití polynomického kernelu SVM nebylo dosaženo uspokojivých výsledků. V tabulce 3 lze pozorovat míru úspěšnosti vzhledem k počtu natrénovaných subjektů.

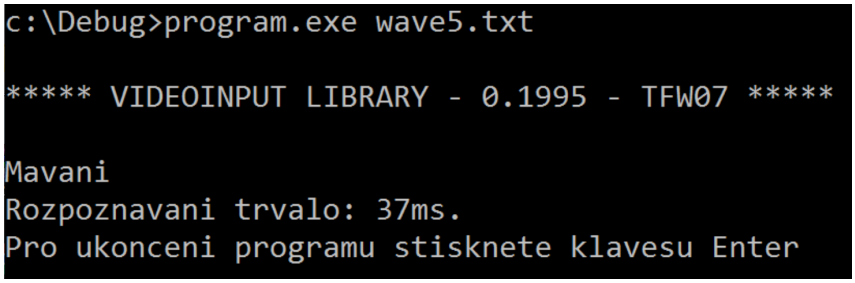
Rozdělením trajektorie na méně než 5 částí byla trajektorie pohybu příliš zobecněna, tudíž toto nevedlo k uspokojivým výsledkům.



Obrázek 29: Porovnání úspěšnosti rozpoznávání akcí při rozdělení trajektorie na 5 částí a užití lineárního a polynomického kernelu SVM. Svislá osa znázorňuje úspěšnost v procentech, vodorovná počet natrénovaných subjektů.

Tabulka 4: Rychlost algoritmu vzhledem k počtu snímků v sekvenci

Počet snímků	Čas [ms]
11	20
16	27
23	39
39	65
78	124



Obrázek 30: Příklad použití aplikace.

4.6 Finální aplikace

Finální aplikace na rozpoznávání a klasifikaci akcí využívá rozdělení trajektorie na 5 částí, užití lineárního kernelu SVM a 4 natrénovaných subjektů. Toto rozhodnutí bylo učiněno na základě testování, kdy právě s tímto nastavením SVM bylo dosahováno nejvyšší míry úspěšnosti rozpoznávání akcí. Pro běh aplikace je vyžadován 64bitový operační systém Windows a nainstalovaná knihovna OpenCV ve verzi 3.1.

Aplikace se spouští z příkazového řádku systému Windows a přijímá jeden parametr ve formě textového souboru z datasetu. Výstupem aplikace je slovní pojmenování rozpoznané akce a informace o délce běhu algoritmu.

5 Závěr

Cílem této bakalářské práce bylo seznámení se základními metodami rozpoznávání akcí v obraze. Byly popsány možnosti snímání obrazu pro účely rozpoznávání akcí, techniky detekce pohybu ve videosekvencích, druhy prostorových reprezentací akcí, metody získávání příznaků z obrazu a metody strojového učení. Dále byl implementován algoritmus rozpoznávání akcí prováděných člověkem založených na trajektoriích pohybu jednotlivých kloubů. Pro implementaci byl zvolen programovací jazyk C++ z důvodu jeho rozšířenosti a multiplatformnosti. Ze stejného důvodu byla zvolena podpůrná knihovna pro práci s obrazem OpenCV.

V průběhu testování vyvinutého algoritmu bylo zjištěno, že na správnost rozpoznávání má velký vliv použitý kernel SVM klasifikátoru a počet trénovacích dat. Jako ideální výsledek se ukázalo použití lineárního klasifikátoru a natrénování čtyř subjektů pro každou třídu akcí. Při této konfiguraci byl algoritmus schopen dosahovat úspěšnosti rozpoznávání prováděných akcí 83,34%.

Algoritmus může být dále rozšířen o SVM klasifikátor typu ONE_CLASS, který by zajistil, že vstupní data neznámých tříd nebudou dále zpracovávána. V rámci testování rychlosti provádění algoritmu bylo také zjištěno, že na běžném laptopu je akce o 30 snímcích vyhodnocena v průměru za 50ms. Na základě tohoto zjištění lze předpokládat, že algoritmus je schopen pracovat také v reálném čase.

Vývoj tohoto algoritmu prověřil a dále zdokonalil mé programovací znalosti v jazyce C++. Blíže jsem se seznámil s knihovnou OpenCV, dozvěděl se spoustu informací, jak pracovat s obrazem. S ohledem na absenci tuzemských zdrojů literatury, k některým částem práce, jsem byl nucen využít literaturu cizojazyčnou, čímž jsem získal cenné zkušenosti.

Literatura

[1] Princip snímání a záznamu obrazu. Populárně naučný portál Popular [online]. [cit. 2017-04-21]. Dostupné z: http://popular.fbmi.cvut.cz/optoel/Stranky/Princip-sn%C3%ADm%C3%A1n%C3%AD-a-z%C3%A1znamu-obrazu.aspx

[2] Jungong HAN, Ling SHAO a Dong XU. Enhanced Computer Vision with Microsoft Kinect Sensor: A Review [online]. [cit. 2017-04-21]. Dostupné z: https://www.microsoft.com/en-us/research/publication/enhanced-computer-vision-with-microsoft-kinect-sensor-a-review/

[3] Detekce pohybu ve videu a jejich identifikace [online]. [cit. 2017-04-21]. Dostupné z: http://fbmi.cvut.cz/files/predmety/3528/public/Detekce%20pohybu%20ve%20videu.pdf

[4] NC State University [online]. [cit. 2017-04-21]. Dostupné z: http://www4.ncsu.edu/ kay-/msf/fig2629.jpg

[5] Common Vision Blox [online]. [cit. 2017-04-21]. Dostupné z: https://www.commonvisionblox.com/media/uploads/products/software/CVB/CVB-Optical-Flow-App1-I0.jpg

[6] Daniel WEINLAND, Remi RONFARD a Edmond BOYER. A Survey of Vision-Based Methods for Action Representation, Segmentation and Recognition [online]. [cit. 2017-04-21]. Dostupné z: https://hal.inria.fr/hal-00640088/file/weinland10preprint.pdf

[7] University at Buffalo [online]. [cit. 2017-04-21]. Dostupné z: http://wings.buffalo.edu/research/cubs/images/truthing.jpg

[8] Tracking Users with Kinect Skeletal Tracking. MSDN home [online]. [cit. 2017-04-21]. Dostupné z: https://msdn.microsoft.com/en-us/library/jj131025.aspx

[9] Ivan LAPTEV. Human action recognition - Ivan Laptev. SlideShare [online]. [cit. 2017-04-21]. Dostupné z: https://image.slidesharecdn.com/cvml2011ivan-110826013334-phpapp02/95/cvml2011-human-action-recognition-ivan-laptev-66-728.jpg?cb=1314322899

[10] Alireza Shafaei [online]. [cit. 2017-04-21]. Dostupné z: http://www.cs.ubc.ca/s̃hafaei/homepage/projects/images/bmvc14_1.jpg

[11] Mohammad HAGHIGHAT, Mohamed ABDEL-MOTTALEB a Wadee S. ALHALABI. Fully Automatic Face Normalization and Single Sample Face Recognition in Unconstrained Environments [online]. [cit. 2017-04-21]. Dostupné z: https://www.researchgate.net/publication/284243500_Fully_Automatic_Face _Normalization_and_Single_Sample_Face_Recognition_in_Unconstrained_Environments

[12] Jakub HARTMANN. Příznakové rozpoznávání [online]. 2012 [cit. 2017-04-21]. Dostupné z: http://dspace.vsb.cz/handle/10084/93101

[13] Local Feature Detection and Extraction. MathWorks [online]. [cit. 2017-04-21]. Dostupné z: https://www.mathworks.com/help/vision/ug/local-feature-detection-and-extraction.html

[14] Navneet DALAL a Bill TRIGGS. Histograms of Oriented Gradients for Human Detection [online]. [cit. 2017-04-21]. Dostupné z: https://hal.inria.fr/file/index/docid/548512/filename/hog_cvpr2005.pdf

[15] Pattern recognition systems – Lab 5: Histograms of Oriented Gradients [online]. [cit. 2017-04-21]. Dostupné z: http://users.utcluj.ro/řaluca/prs/prs_lab_05e.pdf

[16] Ivan LAPTEV a Tony LINDEBERG. Space-time Interest Points [online]. [cit. 2017-04-21]. Dostupné z: http://www.irisa.fr/vista/Papers/2003_iccv _laptev.pd

[17] Spatio - temporal features [online]. [cit. 2017-04-21]. Dostupné z: http://www.micc.unifi.it/seidenari/wp-content/uploads/2010/01/A51-Spatio-temporal-features1.pdf

[18] Antonio W. VIEIRA, Gabriel L. OLIVEIRA, Erickson R. NASCIMENTO, Zicheng LIU a Mario M. CAMPOS. STOP: Space-Time Occupancy Patterns for 3D Action Recognition from Depth Map Sequences [online]. [cit. 2017-04-21]. Dostupné z: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.724.8106&rep=rep1&type=pdf

[19] WANG, Jiang, Zicheng LIU a Ying WU. Human Action Recognition with Depth Cameras. Aufl. 2014. Cham: Springer International Publishing, 2014. ISBN 978-331-9045-610.

[20] Imran N. JUNEJO, Emilie DEXTER, Ivan LAPTEV a Patrick PÉREZ. View-Independent Action Recognition from Temporal Self-Similarities [online]. [cit. 2017-04-21]. Dostupné z: http://www.irisa.fr/vista/Papers/2010_pami_junejo.pdf

[21] An Introduction To Machine Learning. Fewer Lacunae [online]. [cit. 2017-04-21]. Dostupné z: https://kevinbinz.com/2014/08/23/an-introduction-to-machine-learning/

[22] Machine Learning, Part I: Supervised and Unsupervised Learning. Aihorizon [online]. [cit. 2017-04-21]. Dostupné z: http://www.aihorizon.com/essays/generalai/supervised_unsupervised_machine_learning.htm

[23] Zdeněk DOSTÁL a Vít VONDRÁK. LINEÁRNÍ ALGEBRA [online]. [cit. 2017-04-21]. Dostupné z: http://mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/linearni_algebra.pdf

[24] Stephen PRATA. Mistrovství v C. 4., aktualiz. vyd. Brno: Computer Press, 2013. Bestseller (Computer Press). ISBN 978-80-251-3828-1.

[25] John MACCORMICK. Https://users.dickinson.edu/ jmac/selected-talks/kinect.pdf [online]. [cit. 2017-04-21]. Dostupné z: https://users.dickinson.edu/j̃mac/selected-talks/kinect.pdf

[26] Kinect. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-04-21]. Dostupné z: https://en.wikipedia.org/wiki/Kinect

[27] Xia, L. and Chen, C.C. and Aggarwal. JK View invariant human action recognition using histograms of 3D joints. UTKinect-Action3D Dataset [online]. [cit. 2017-04-21]. Dostupné z: http://cvrc.ece.utexas.edu/KinectDatasets/HOJ3D.html

[28] OpenCV [online]. [cit. 2017-04-21]. Dostupné z: http://opencv.org/

[29] Jamie SHOTTON, Andrew FITZGIBBON, Andrew BLAKE, Alex KIPMAN, Mark FINOCCHIO, Bob MOORE a Toby SHARP. Real-Time Human Pose Recognition in Parts from a Single Depth Image [online]. [cit. 2017-04-21]. Dostupné z: https://www.microsoft.com/en-us/research/publication/real-time-human-pose-recognition-in-parts-from-a-single-depth-image/

[30] Lagrangian trajectory. ENS [online]. [cit. 2017-04-21]. Dostupné z: http://perso.ens-lyon.fr/emmanuel.leveque/Lagrangian_trajectory.png

[31] LIBSVM – A Library for Support Vector Machines [online]. [cit. 2017-04-21]. Dostupné z: https://www.csie.ntu.edu.tw/čjlin/libsvm/

[32] F1 score. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-04-24]. Dostupné z: https://en.wikipedia.org/wiki/F1_score

A Struktura příloh na CD

- dataset.zip
- program.exe
- SVM.xml
- VS_project.zip
- thesis.pdf